JP62094222

Publication Title:

Thread cutting control method

Abstract:

There is provided a thread cutting control method in an NC apparatus having a CPU in an arithmetic unit whereby real time clocks which are generated for every constant time are received as an interruption signal and a sampling operation is performed in response to the interruption signal, thereby controlling the feed system of a machine tool having a thread cutting function. In this system, the rotational speed of a spindle which is rotating asynchronously with the real time clocks is measured; the rotational amount of the spindle from a predetermined position on the rotational position of the spindle until the real time clock is just generated is measured; and after a feed system adapted to perform a thread cutting work on the basis of the rotational amount and the rotational speed has moved by a predetermined distance, a feed command to control the feed system is generated such that the movement distance is constant for the position of the spindle and at the same time, a predetermined ratio is held between the feed speed of the feed system and the rotational speed of the spindle. With this system, the feed shaft to cut the thread lead can be controlled with a high degree of accuracy.

---

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭62－94222

⑤1Int.Cl.⁴	識別記号	庁内整理番号	④3公開　昭和62年(1987)4月30日
B 23 G 1/02		A－7041－3C	

審査請求　未請求　発明の数 1　（全8頁）

⑤4発明の名称　ネジ切削制御方式

⑵1特　願　昭60－233503
⑵2出　願　昭60(1985)10月21日

⑦2発明者　山中　守　入間市大字上藤沢字下原480番地　株式会社安川電機製作所東京工場内
⑦2発明者　杉村　昌彦　入間市大字上藤沢字下原480番地　株式会社安川電機製作所東京工場内
⑦1出願人　株式会社安川電機製作所　北九州市八幡西区大字藤田2346番地
⑦4代理人　弁理士　若林　忠

明　細　書

1.発明の名称

ネジ切削制御方式

2.特許請求の範囲

(1) 演算部にＣＰＵを有し、一定時間毎のリアルタイムクロックを割込み信号として入力し、該信号によりサンプリング制御を行ない、ネジ切削機能を有する工作機械の送り軸を制御するＮＣ装置において、

リアルタイムクロックとは非同期に回転している主軸の回転速度の測定と、主軸の回転位置のある定められた位置からリアルタイムクロックが発生した瞬間までの主軸の回転量の測定を行ない、その回転量と回転速度をもとに切削を行なう送り系がある一定距離移動した後は、その移動距離が主軸位置に対して一定で、かつ送り速度が主軸の回転速度に対してある定められた比となるように送り軸を制御する送り指令発生を行なうことを特徴とするネジ切削制御方式。

(2) 主軸の回転量および回転速度を主軸またはワークの回転軸に取り付けたパルスジェネレータからのフィードバックパルスをカウントすることにより行なう特許請求の範囲第1項記載のネジ切削制御方式。

(3) サンプリング制御をソフトウエアにより行なう特許請求の範囲第1項記載のネジ切削制御方式。

(4) ネジ切削スタート時のサンプリング周期のみ前記回転量にある定数を乗じ、スタート以降のサンプリング周期では前記回転速度に前記定数を乗ずることにより所望の送り速度発生を行なう特許請求の範囲第1項記載のネジ切削制御方式。

(5) 前記回転量からある値を減算し、その結果を前記回転量に見立てて送り速度発生を行なう特許請求の範囲第1項記載のネジ切削制御方式。

3.発明の詳細な説明

（産業上の利用分野）

本発明はネジ切削制御装置、特に数値制御されるネジ切削制御装置におけるネジ切削制御方式に関する。

（従来の技術）

第９図は旋盤におけるネジ切りの概念図である。

ワーク42の切削のための刃物41は送り軸として制御される。ここで、送り軸とはＮＣ装置により位置、速度を制御されるもので、刃物台を動かすモータ、そのモータを駆動するサーボアンプ、モータの回転位置を検出するパルスジェネレータ、モータの速度を検出するタコジェネレータ、サーボアンプへ指令を与えるＮＣ装置等からなる制御系を言う。ワーク42は主軸モータにより回転させられ、このワーク42の表面に刃物41を第９図矢印方向へ送ることによりネジ切りが行なわれる。主軸ＰＧ（以下、ＳＰＧと称す）43はワーク42の回転を検出するもので、１回転毎に原点パルスを出力する。このＳＰＧ43は一般には、4096パルス／１回転程度の分解能があり、このＳＰＧ43のフィードバック（以下、Ｆ.Ｂと称す）パルスによりＮＣ装置で刃物台の送り速度を制御することになる。

ＮＣでは一般に、mm/minで指定する送りとmm/rev（ネジ切りなどの主軸回転に対する送り）で指定する送りと、大きく分けて２種類の送りがある。mm/minを指定したときはスイッチSW_1がオンして時間基準のクロックが速度発生器45へ入力され、mm/revを指定したときはスイッチSW_2がオンしてＳＰＧパルスが速度発生器45へ入力される。速度発生器45は、これらの入力を基準にして送り軸の実際の速度の基本となるパルス53を発生する。関数発生器46はパルス53を基に送り軸の指令パルス54を発生する。一般には２軸またはそれ以上が同期して動くので、この関数発生器46で複数軸への指令パルスを同時に発生する。偏差カウンタ47は関数発生器46で発生した指令パルス54を加算して送り軸のＰＧ52のＦＢパルス55を減算し、指令した位置と現在位置の偏差をとる。Ｄ／Ａ変換器48は偏差カウンタ47の偏差をデジタル／アナログ変換し、サーボアンプ49にアナログ信号の指令として出力する。サーボアンプ49はこの指令に基づいてモータ50を駆動する。つまり、偏差カウンタ47、Ｄ／Ａ変換器48、サーボアンプ49、モータ50、ＰＧ52の一巡するループで位置制御ループが構成されている。ここでは、モータ軸に取り付けられた速度検出器（ＴＧ）51からのＦＢにより速度制御も行なう。ただし、これはあくまでＤ／Ａ変換器48からの指令どおりの速度でモータ50が回転するように制御系を補償するためのものである。

この方式では、ネジ切りの始めのタイミング合わせのため、原点検出器44でＳＰＧ52の原点パルスを検出し、検出した瞬間にスイッチSW_3をオンさせ速度発生器45のパルス発生をスタートさせている。つまり、ＳＰＧ52の原点に同期して送り軸の指令がスタートすることになる。また、主軸との速度の同期については、速度発生器45の基本クロックをＳＰＧパルスとすることで対処している。これらにより、前述した２つの問題点を解決している。

第11図はソフトウエアサーボ方式の従来例を示すブロック図である。ここでは、送り軸への指令

ネジ切りとは、ワーク42の表面にネジ状のリードを削るものであるが、注意しなければならない問題点が大きく２つある。ネジ切りは１回の送りで終了するものではなく、ネジの溝を深くするためには一般には同じ所を複数回、削る必要がある。そのために、まず第１に毎回ワークの同じ点から切り込まなければならないということである。言い換えると、ネジを切り始める位置はＳＰＧ43の原点位置またはその原点を基準として一定の位置（角度）でなければならない。第２に、ネジのリードのピッチを正確にするため送りの速度をワーク42の回転速度（実際にはＳＰＧ43の回転速度）に比例させることが必要である。この回転速度は、ネジ切りの１回目と２回目以降で異ることもあるし、ネジ切りの途中で主軸の回転速度が変化するということもある。つまり、時々刻々変化しうる主軸の回転に合わせて送り速度を制御する必要がある。

第10図は送り軸の制御方式の従来例を示すブロック図である。

の発生および送り軸の位置制御をマイクロプロセッサによりソフトウエアで制御することをソフトウエアサーボと呼ぶ。ソフトウエア制御ではサンプリング制御となる。つまり、データ処理部56で周期Ts毎に指令発生、位置の偏差量の作成、D／A変換器48へのデータ作成等の演算を行なう。主軸の速度を知るために、SPGパルスをカウントするSPGカウンタ57が設けられており、SPGデータとしてサンプリングされる。また、送り軸の位置もPGパルスをFBカウンタ58でカウントしてFBデータとしてサンプリングされる。なお、D／A変換器48、サーボアンプ49、モータ50、TG51、SPG52については第10図の従来方式と同じものである。

従来方式とソフトウエアサーボ方式の大きな違いは、前者がパルス単位を基本としたハードウエアで構成されているのに対し、後者は、パルス（SPGも送り軸のFBPGも）はカウンタ57，58にためて一定周期でデータとしてサンプリングし、その後の処理は全てソフトウエアによりデータとして演算されているということである。

現在、工作機械は高精度、高速化される傾向にある。高精度化とは検出単位を1μmから0.1μmにするというような意味である。ここにパルスを基本単位とした従来方式での限界が発生する。例えば、1パルス＝1μmで送り速度F＝24m/minとすると、PPS(パルス／sec)は400 KPPSとなる。この程度のPPSであればNC内部のハードウエアは十分に対応できる。しかし、1パルス＝0.1 μmで送り速度F＝24m/minとなると4 MPPSと10倍高速になる。そうなると、1パルス毎に動作するハードウエアではこの速度に対応できなくなる箇所が発生してくる。ところが、データとして処理するソフトウエアサーボでは、この場合データ長がのびるというだけで、演算処理はあくまで定められたクロック（例えば1 msecとか2 ms）毎に行なえばよい。つまり、ソフトウエアサーボは高速、高精度向きといえる。さらに、マイクロプロセッサ等の進歩、低価格化も考え合わせれば、ソフトウエアサーボ方式により従来方式よりも低価格で高機能のNC装置が実現できる。

（発明が解決しようとする問題点）

このように、ソフトウエアサーボ方式は利点が多いものの、欠点もある。その1つがネジ切りの問題である。

ソフトウエアサーボでは、サンプリング制御であるため、サンプリング周期Tsの間で発生した事象に対してはデータ処理部がこれを認知するのに最大1サンプル時間、さらに演算処理して指令をD／A変換器に出力するのに1サンプル時間を要する。つまり、2サンプル時間の遅れが生じることになる。また、その事象がサンプル時間Tsの中のどの時点で発生したかは特別なハードウエアを準備しないかぎり認識できない。

そのため、ネジ切りのようにSPGの原点パルスと同時に送り軸を動かすというような動作が一般にはむずかしくなる。

本発明の目的は、ソフトウエアサーボ方式のNC装置において、精度良くネジ切りを行なわせるネジ切削制御方式を提供することである。

（問題点を解決するための手段）

本発明のネジ切削制御方式は、演算部にCPUを有し、一定時間毎のリアルタイムクロックを割込み信号として入力し、該信号によりサンプリング制御を行ない、ネジ切削機能を有する工作機械の送り軸を制御するNC装置において、リアルタイムクロックとは非同期に回転している主軸の回転速度の測定と、主軸の回転位置のある定められた位置からリアルタイムクロックが発生した瞬間までの主軸の回転量の測定を行ない、その回転量と回転速度をもとに切削を行なう送り系がある一定距離移動した後は、その移動距離が主軸位置に対して一定で、かつ送り速度が主軸の回転速度に対してある定められた比となるように送り軸を制御する送り指令発生を行なうことを特徴とする。

（作　用）

第4図はエアカット部のあるネジ切りの動作を示す図で、ネジ切りをする場合の刃物61と主軸により回転させられるワークの相対的動き65→66→

67→68を示している。図中、63はエアカット部と呼び実際には切削を行なわない。実際に切削しているのは64の部分である。送り軸のサーボ系は一般的に位置ループゲインKpで決まる遅れ系になっており、指定の速度まで加速するのにある時間を要する。そのためネジ切りの場合、精度良くネジを切るためには必ずある距離以上の加速距離が必要である。今、この距離をLとすると、これはエアカット部63の距離に等しい。前述したようにネジ切りの時は刃物61がワーク62に切り込む時、つまり64の部分では主軸の回転に対して常に一定の関係（当然、速度も一定の関係）でなければならない。送り軸はＲＴＣ（リアルタイムクロック）に必ず同期して移動し始めるので、その時の主軸の位置（角度）は一定ではない。そこで、加速距離Lを動く間で主軸の回転に対して、相対位置がうまく一定になるように加速の仕方を制御すればよいことになる。

第５図はネジ切り等の加速の様子を示すグラフである。時刻t_0はネジ切りがスタートしたＲＴＣのタイミングであり、τ_1, τ_2, τ_3は主軸原点パルスが発生した時点とＲＴＣの時間的ずれである。f_1, f_2, f_3は送り軸の速度カーブであり、それぞれ主軸原点パルスとＲＴＣの時間的ずれτ_1, τ_2, τ_3に対応している。l_1, l_2, l_3はそれぞれ速度カーブf_1, f_2, f_3の場合の送り軸の移動距離がエアカット距離Lになるまでの時間であり、以下の式が成立する。

$$L = \int_0^{l_1} f_1(l)\,dl = \int_0^{l_2} f_2(l)\,dl = \int_0^{l_3} f_3(l)\,dl \quad \cdots (1)$$

また、このときの送り軸の速度も等しくなければならないので、

$$v = f_1(l_1) = f_2(l_2) = f_3(l_3) \quad \cdots (2)$$

ただし、Ｖはネジ切り中の送り速度

つまり、主軸原点とネジ切りをスタートするときのＲＴＣとの時間的ずれτに対して (1)式と(2)式が成立するように速度カーブｆを選べば精度良くネジが切れる。

以上は時間を基準に述べたが、主軸位置Ｓを基準にして示すと第６図のようになる。ここで、主軸位置Ｓはネジ切りをスタートしたときの主軸原点からの主軸ＦＢパルスの積算値である。S_0はネジを切り始める主軸（ワーク）位置である。S_1, S_2, S_3は主軸原点発生後のＲＴＣで送り軸がスタートしたときの主軸位置である。この$S_1 \sim S_3$の値はいろいろな値にバラつく。この値に対し、以下の式が成立するように送り指令発生の関数g_1, g_2, g_3を選べばよい。

$$L = \int_{S_1}^{S_0} g_1(S)\,ds = \int_{S_2}^{S_0} g_2(S)\,ds = \int_{S_3}^{S_0} g_3(S)\,ds \quad \cdots (3)$$

$$V \cdot g_1(S_0) = g_2(S_0) = g_3(S_0) \quad \cdots (4)$$

これら(3)，(4)式が成立するような主軸速度Ｓに対する送り指令の関数ｇの最も簡単な例を第７図に示す。図中11_1，11_2，…，11_5はＲＴＣを表わしている。この例ではＲＴＣの11_1と11_2間で主軸原点パルスが発生している。S_1, S_2, S_3, S_4はこれらＲＴＣ間の一定時間に主軸が回転した量（単位は主軸ＦＢパルス）であり、主軸速度を意味している。ここで、S_2以後はＲＴＣ１周期間の主軸回転量であるが、S_1のみは主軸原点からＲＴＣまでの間の主軸回転量である。ｖは単位主軸速度当りの送り軸速度であり、ＣＰＵによりｖ×Ｓの乗算をして送り指令とする。つまり、指定データの$v \times S_1 \rightarrow v \times S_2 \rightarrow v \times S_3 \rightarrow \cdots$というＲＴＣ毎に階段状に変化する関数が関数ｇに他ならない。以下、簡単にその証明を行なう。

指令$v \times S_4$の途中で所定の（切り込むべき）主軸位置S_0になったとすると、

$$S_0 = S_1 + S_2 + S_3 + \delta \cdot S_4 \quad \cdots\cdots (5)$$

ただし、δは指令$v \times S_4$の間にS_0に達したことを意味しており、$\delta = 0 \sim 1$である。

このときの送り軸の移動量ℓは

$$\ell = v \times S_1 + v \times S_2 + v \times S_3 + v \times \delta \times S_4$$
$$= v \times S_0 \quad \cdots\cdots (6)$$

つまり、移動量ℓは主軸位置S_0に対し一定である。また、送り指令は$v \times Si$であるので常に主軸速度に対し一定の比になっている。

なお、第８図に示すように、主軸原点から意図的にある量（ある角度）だけずらしてネジ切りを

スタートさせてもよい、この場合、ＣＰＵはＲＴＣの処理の中で主軸位置S_{RTC}を見てSd以上になったら$S_1 = S_{RTC} - Sd$として前述したのと同様な制御を行なえばよい。これは、あるネジのリード（既に切ったネジのリード）に対して任意の角度、ずらした位置からさらに別のネジのリードを切削することができることを示しており、多条ネジなどの切削に応用できる。

（実施例）

本発明の実施例について図面を参照して説明する。

第１図は本発明のネジ切削制御方式の一実施例を示すブロック図、第２図はＲＴＣ11、ＣＰＵの演算、主軸原点パルスのタイムチャートである。

ソフトウエアサーボではサンプリングにより送り軸の制御を行なう。このサンプリングのクロックとなるＲＴＣ11がＲＴＣ発生器２から発生させられる。演算部１はＣＰＵ、ＲＯＭ、ＲＡＭ等を含む。ＣＰＵはＲＴＣ11を割込み信号としてその割込み処理により制御のための演算を行なう。第２図はその様子を示しており、$RTC11_1$、11_2、…に対してそれぞれ時間T_1、T_2、…に演算が行なわれている。

送り軸の制御の演算は大きく指令データの作成（速度発生、関数発生等）の部分と位置ループ制御の部分に分かれる。

（指令データの作成）

前述したように、軸の送りにはmm/minの送りとmm/revの送りがある。mm/minの送りの場合、時間当りの送り速度が決まっているので、Ts時間毎の移動量に計算し直せばよい。また、mm/revの送りは主軸回転速度との比なので、主軸の速度を知らなければならない。このために、ＳＰＧカウンタ３とＳＰＧラッチ回路４がある。ＳＰＧラッチ回路４はＳＰＧカウンタ３の値をＲＴＣ11のタイミングでラッチする。そして、このラッチされた値をＣＰＵが読み、前回のＲＴＣ11のタイミングで読んだデータからの増分をとることにより主軸速度が正確に測定できる。こうして得た主軸速度（時間Ts当りの主軸回転量）に対し、mm/revで指定された値（実際には単位主軸速度当りの送り量）を乗算すれば、時間Ts当りの送り量になる。実際には、２軸同時補間等の軌跡指令の計算もこの演算部１で行なわれるが、本発明とは直接、関係ないのでその説明は省略する。

（位置ループ制御）

前記指令データ（時間Ts当りの送り量）を積算した値（指令位置データと呼ぶ）と実際にモータが回転した位置（ＦＢ位置）との差を計算し、その値に比例したデータをＤ／Ａ変換器８に与え、サーボアンプへの指令とする。ＦＢ位置は以下のようにして求める。まず、送り軸のＦＢパルス（ＰＧ等による）16を、正転／逆転に合わせてアップ／ダウンするアップ／ダウンカウンタであるＦＢカウンタ９でカウントして、その値をＲＴＣ11のタイミングでＦＢラッチ回路10にラッチする。その後、ＣＰＵにて、前回のＲＴＣ11でのＦＢラッチ回路10のデータからの増分をとれば、前回のＲＴＣ11から今回のＲＴＣ11までの間にモータが回転した量がわかる。さらにそのデータを積算すればＦＢ位置となる。このようにしてＣＰＵで位置ループ制御の演算を行なった結果は一度、バッファ６にセットされ、次のＲＴＣ11のタイミングでラッチ回路７にラッチされてＤ／Ａ変換器８へ入力される。なお、フリップフロップ５は主軸原点検出用で、主軸原点検出パルス14を検出すると、出力がハイレベルになる。

第３図は本実施例における演算処理の流れを示すフローチャートである。

ステップ21→22→23→24は通常の処理で、ネジ切り以外の時やネジ切りがスタートした後のネジ切り中のＲＴＣ11による処理である。フリップフロップ５を強制的にクリアし（ステップ22）、指令値の計算（（指令データの作成）の項で述べた内容）を行ない（ステップ23）、位置ループの制御（（位置ループ制御）の項で述べた内容）を行なう（ステップ24）。

ステップ21→25→26→29→30はネジ切りのスタート待ちの状態の処理である。主軸原点パルスが発生したときに、その発生を検出するためにフ

リップフロップ5のクリアを解除してイネーブルにしている（ステップ25）。そしてネジ切りスタート待ちで停止しなければならないので、送り指令を0にしている（ステップ29）。いわゆるサーボロックの状態である。最後に、ステップ24と同じく位置ループの制御を行なう（ステップ30）。主軸原点パルスが発生するまでは以上のステップを繰り返す。

ステップ21→25→26→27→28は主軸原点パルスが発生した直後、つまりネジ切りスタート時の処理である。ただし、指令値演算（ステップ27）はステップ23の指令値演算と少し異なる。それは、ステップ23の演算では主軸速度は前回のＲＴＣ11からの増分値としているが、ステップ27では主軸速度を主軸原点からのそのＲＴＣ11までの移動量（第７図のS_1）としている点である。それ以外については同じである。位置ループ制御（ステップ28）は、ステップ24，30と同じである。

以上により第７図に示したような指令データにより、送り軸をＲＴＣ11のサンプリングにより制御する。

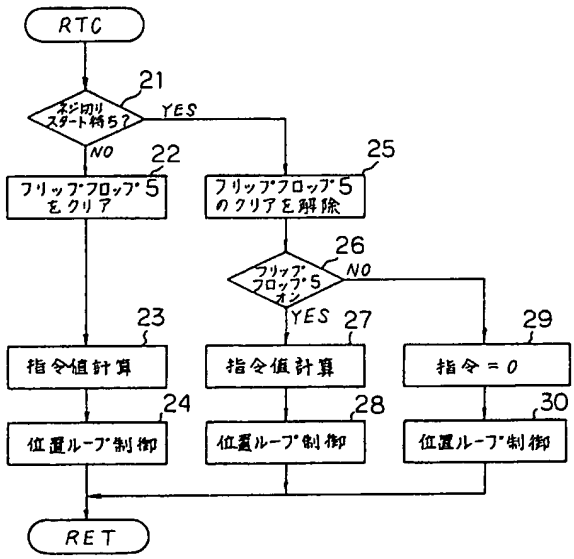
（発明の効果）

以上説明したように本発明は、一定時間間隔で発生するＲＴＣをサンプリングクロックとするＮＣ装置でネジ切りをおこなう際、そのＲＴＣとは非同期に回転する主軸（またはワーク）に対し、常に一定の主軸位置（ワーク位置）から切り込み、かつ主軸の回転速度に対し一定の比の送り速度で送り軸を制御することにより、精度良くネジのリードを切削するための送り軸の制御が可能となる効果がある。

4.図面の簡単な説明

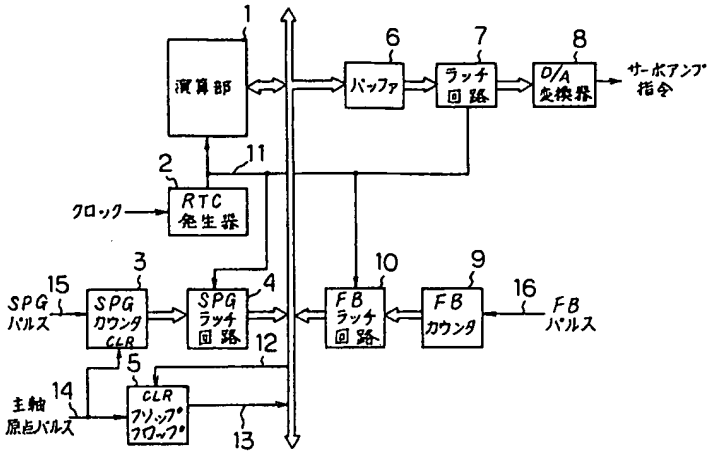
第１図は本発明のネジ切削制御方式の一実施例を示すブロック図、第２図はＲＴＣ11，ＣＰＵの演算、主軸原点パルスのタイムチャート、第３図は第１図の実施例における演算処理のフローチャート、第４図はエアカット部のあるネジ切りの動作を示す図、第５図、第６図は本発明におけるネジ切り時の加速の様子を示す図、第７図、第８図は本発明における送り指令発生の例を示すタイムチャート、第９図はネジ切りの概念図、第10図は従来方式の送り軸制御系のブロック図、第11図はソフトウエアサーボ方式での送り軸制御系のブロック図である。

1…演算部、　2…ＲＴＣ発生器、
3…ＳＰＧカウンタ、　4…ＳＰＧラッチ回路、
5…フリップフロップ、
6…バッファ、　7…ラッチ回路、
8…Ｄ／Ａ変換器、　9…ＦＢカウンタ、
10…ＦＢラッチ回路、11…ＲＴＣ、
12…フリップフロップ５のクリア信号、
13…フリップフロップ５の出力信号、
14…主軸原点パルス、15…ＳＰＧパルス、
16…ＦＢパルス。

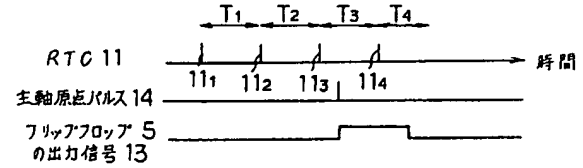


第３図

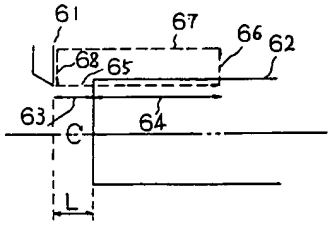
特許出願人　株式会社安川電機製作所
代理人　若林　忠



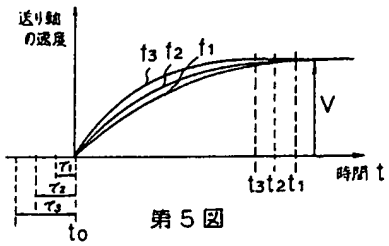
第1図



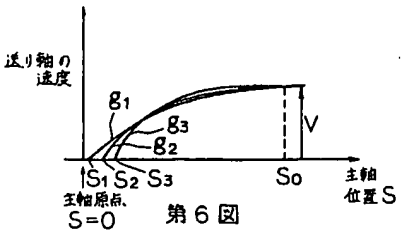
第2図



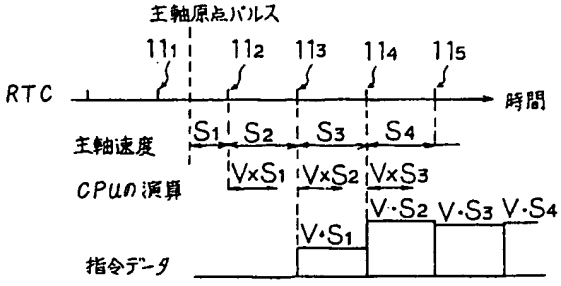
第4図



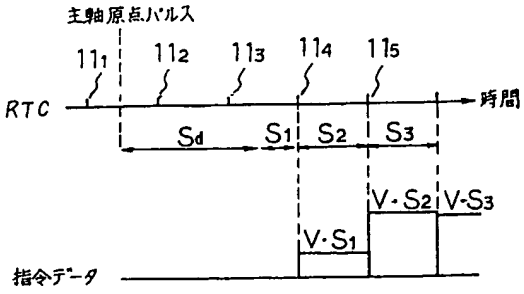
第5図



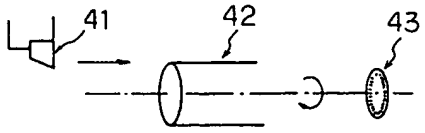
第6図



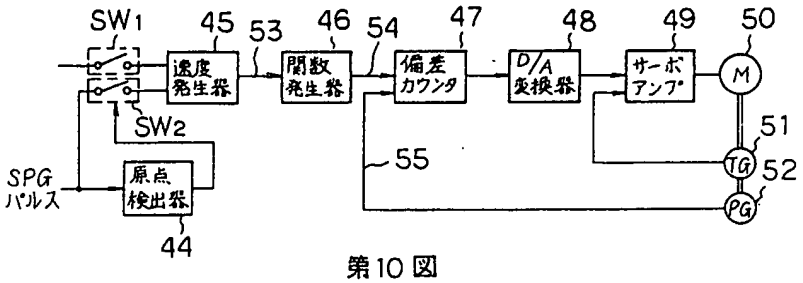
第7図



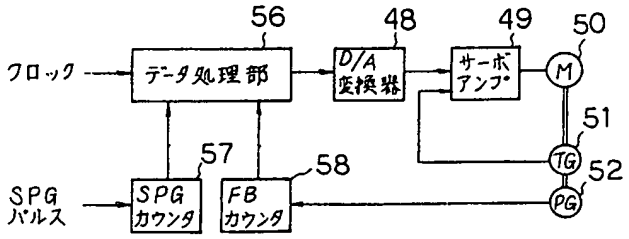
第8図



第9図



第10図



第11図